

# CSED101. Programming & Problem solving

## Spring, 2014

### Programming Assignment #5 (70 points)

성명철(mcs8779@postech.ac.kr)

■ **Due:** 2014.06.13 23:59

■ **Development Environment:** Windows Visual Studio 2010

#### ■ **제출물**

- **C Code files (assn5.c)**
  - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙일 것.
- **보고서 파일** (.doc(x) or .hwp) 예) assn5.doc(x) 또는 assn5.hwp
  - AssnReadMe.pdf 를 참조하여 작성할 것.
- 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

#### ■ **주의사항**

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
- 각 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
- 과제를 작성하는데 있어서 전역변수를 선언하여 사용할 수 없다.
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없다.
- 컴파일 & 실행이 안되면 무조건 0점 처리된다.
- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)

## Problem: 성적 관리 프로그램

### (문제)

파일로부터 데이터를 읽어서 성적 목록을 만들어 관리하는 성적 관리 프로그램을 만들어보자. 이 성적 관리 프로그램에는 성적 관리와 관련된 기능과 그 외의 기능으로 전화번호를 이용한 검색이 포함되어 있다.

### (주의사항)

- 최소한 각 명령어 별로 함수를 정의하여 사용한다. 즉, 명령어 외에 필요한 함수는 정의하여 사용할 수 있다.
- string.h 에서 제공하는 라이브러리 함수를 사용할 수 있다.

### (설명)

➤ 프로그램 구현은 구조체와 linked list를 사용하여야 하며, 각 항목에 대한 제약은 다음과 같다.

- 이름: 공백이 없는 한글 또는 영어 string(한글 최대 5글자, 영어 최대 10글자)
    - ◇ 한글 문자 1개는 2byte의 저장공간이 필요하다.
    - ◇ 이름은 식별자(identifier)로 동일한 이름을 가진 경우는 없다.
  - 전화번호: 최대 11개의 숫자로만 이루어진 string
  - 성별: '남' 또는 '여' (한글 1글자)
  - 중간고사 점수: 0~100까지의 값을 가지는 integer
  - 기말고사 점수: 0~100까지의 값을 가지는 integer
  - 성적(grade): 1글자 character ('A', 'B' 등)
- ※ 이름, 전화번호 항목에 해당되는 문자열은 공백이 없다고 가정한다.

➤ 프로그램 실행은 다음과 같이 한다. (노란색으로 표시된 문자는 사용자 입력에 해당)

```
C:\W>assn5.exe students.txt
```

- 파일명을 입력하지 않을 경우, default로 "students.txt"로부터 데이터를 읽는다.
- 파일명이 입력될 경우, 입력된 파일로부터 데이터를 읽는다.
- 파일명의 최대길이는 20자이며 파일명에는 공백이 없다고 가정한다.

- ▶ 프로그램 실행 시 텍스트 파일로부터 학생들의 성적 목록 작성을 위한 데이터를 읽으며, 텍스트 파일의 내용 및 구성은 아래와 같다.

홍길동	01012345678	남	84	73
Alice	0541236543	여	92	89
김철수	01098761234	남	57	62
이영희	021112222	여	81	84
Bob	0312254567	남	58	68

- 이름, 전화번호, 성별, 중간고사 점수, 기말고사 점수로 구성되어 있으며 각 항목 사이는 **tab(Wt)**으로 구분된다.
- 학생과 학생 사이는 **줄 바꿈 문자(Wn)**으로 구분된다.
- Ex. [Name][Wt][PhoneNumber][Wt][Gender][Wt][Mid][Wt][Final][Wn]
- 프로그램을 실행시키면 텍스트 파일로부터 데이터를 읽어 목록을 linked list에 저장하고, 전체 목록을 **이름을 기준**으로 정렬(아래 정렬 기준 참고)하여 아래의 예제처럼 출력한다.

※ 정렬 기준

- ① 영어->한글 순
- ② 영어는 알파벳 순, 한글은 가나다 순
- 성적(Grade) 항목은 중간고사 점수와 기말고사 점수의 평균을 계산하여 저장한다. Grade의 기준은 아래와 같다.
  - A: 평균이 90 점 이상
  - B: 평균이 80 점 이상, 90 점 미만
  - C: 평균이 70 점 이상, 80 점 미만
  - D: 평균이 60 점 이상, 70 점 미만
  - F: 평균이 60 점 미만
- 전체 목록을 출력한 후에는 명령어를 입력 받을 준비를 한다.

```

목록
      Name      : Phone Number : Gender : Midterm : Final : Grade
1. Alice      : 0541236543 : 여     : 92      : 89    : A
2. Bob        : 0312254567 : 남     : 58      : 68    : D
3. 김철수     : 01098761234 : 남     : 57      : 62    : F
4. 이영희     : 021112222  : 여     : 81      : 84    : B
5. 홍길동     : 01012345678 : 남     : 84      : 73    : C
Command:
  
```

- 예외처리:
  - ✓ 파일명에 해당하는 파일이 없을 경우 아래와 같은 에러 메시지를 출력하고 프로그램을 종료한다.

```

Cannot open text file. Program exited.
계속하려면 아무 키나 누르십시오 . . .
  
```

## (기능)

- 성적 관리 프로그램은 아래와 같은 기능을 가진다.
- 명시된 8가지 명령어 외의 명령어가 입력될 경우 무시하고 다시 명령어 입력을 대기한다.

### 1. 전체 목록 보기 (show)

- `show` 입력 시, 저장되어 있는 전체 목록을 이름을 기준으로 영어(알파벳 순) -> 한글(가나다 순) 순으로 출력해준다.

```
Command: show
목록

      Name      : Phone Number : Gender : Midterm : Final : Grade
1. Alice       : 0541236543 : 여     : 92      : 89    : A
2. Bob         : 0312254567 : 남     : 58      : 68    : D
3. 김철수      : 01098761234 : 남     : 57      : 62    : F
4. 이영희      : 021112222  : 여     : 81      : 84    : B
5. 홍길동      : 01012345678 : 남     : 84      : 73    : C

Command: _
```

### 2. 이름 찾기 (findname)

- `findname` 입력 시, 찾고자 하는 학생의 이름을 입력 받은 후, 그 학생이 목록에 존재하면 찾아서 아래와 같이 출력한다.
- 예외처리:
  - ✓ 주어진 이름을 가진 학생이 존재하지 않는 경우에는 아래와 같이 메시지 "No results."를 출력한다.

```
Command: findname
Name to search: John → 결과가 없는 경우
No results.

Command: findname
Name to search: 김철수
Name      : Phone Number : Gender : Midterm : Final : Grade
김철수    : 01098761234  : 남     : 57      : 62    : F

Command: _
```

### 3. 부분 전화번호로 검색 (searchnum)

- **searchnum** 입력 시, 찾고자 하는 전화번호의 일부(subNum)를 입력 받아, 목록에 저장된 학생 중 이 subNum 을 포함하고 있는 전화번호를 가진 학생들을 모두 출력한다.
- subNum 의 길이는 3~7 개의 숫자로 제한. 즉, 3~7 개 사이의 입력만 들어온다고 가정한다. (ex. 12345678(x), 12(x))
- 힌트: string.h 에 있는 함수 중 strstr 함수 사용
- 예외처리:
  - ✓ subNum 을 포함하는 전화번호가 존재하지 않으면 아래와 같은 메시지 "No results." 출력

```
Command: searchnum
Number to search: 246 → 결과가 없는 경우
No results.

Command: searchnum
Number to search: 4567
Name      | Phone Number | Gender | Midterm | Final | Grade
Bob       | 0312254567   | 남     | 58      | 68    | D
홍길동   | 01012345678  | 남     | 84      | 73    | C

Command: 
```

### 4. 점수 수정 (changescore)

- 목록에 저장된 학생 중 1 명의 중간고사(*mid*) 혹은 기말고사(*final*)의 점수를 수정한다.
- **changescore** 입력 시, 수정하고자 하는 학생의 이름, 수정하고자 하는 점수가 중간고사인지 기말고사인지와 수정하고자 하는 점수를 순서대로 입력 받아 해당 학생의 점수를 수정한다.
- 점수가 바뀔에 따라 Grade 도 다시 계산하여 수정한다.
- 예외처리: 다음과 같은 경우에는 실행되지 않음
  - ✓ 이름이 목록에 없는 경우
  - ✓ "mid" 또는 "final" 외의 값이 입력된 경우
  - ✓ 점수에 0~100 외의 값이 입력된 경우

```
Command: changescore
Name: John → 이름이 없는 경우

Command: changescore
Name: Alice
Mid/Final?: miid → mid/final 외의 값이 입력된 경우

Command: changescore
Name: Alice
Mid/Final?: mid
Input new score: 147 → 0~100 외의 값이 입력된 경우

Command: changescore
Name: Alice
Mid/Final?: mid
Input new score: 76
Name      | Phone Number | Gender | Midterm | Final | Grade
Alice     | 0541236543   | 여     | 92      | 89    | A
Score changed.
Alice     | 0541236543   | 여     | 76      | 89    | B

Command: 
```

## 5. Grade 검색 (searchgrade)

- `searchgrade` 입력 시, 특정 grade 를 입력 받아 그 grade 에 해당하는 학생을 모두 출력한다.
- 예외처리:
  - ✓ A, B, C, D, F 외의 값이 입력된 경우 실행되지 않음.
  - ✓ 해당 grade 의 학생이 없는 경우 아래와 같이 메시지 "No results." 출력

```
Command: searchgrade
Grade to search: E → A/B/C/D/F 외의 값이 입력된 경우

Command: searchgrade
Grade to search: A → 해당 grade의 학생이 없는 경우
No results.

Command: searchgrade
Grade to search: B
Name      | Phone Number | Gender | Midterm | Final | Grade
Alice     | 0541236543   | 여     | 76      | 89    | B
이영희    | 021112222   | 여     | 81      | 84    | B

Command: _
```

## 6. 학생 추가 (add)

- `add` 입력 시, 아래 예제와 같이 학생의 이름, 전화번호, 성별, 중간고사 점수, 기말고사 점수를 순서대로 입력 받아 새로운 학생을 추가한다. 추가되면, 메시지 "Student added."를 아래 예제와 같이 출력한다.
- Grade 는 중간고사 점수와 기말고사 점수를 사용하여 계산하여 저장한다.
- 학생 추가 후 `show` 명령어를 사용하면 영어(알파벳 순) -> 한글(가나다 순) 순으로 정렬된 상태를 유지한다.
- 이름은 영어 10 글자(한글 5 글자), 전화번호는 11 개 이하의 숫자로만 이루어진 값이 입력된다고 가정한다.
- 예외처리: 다음과 같은 경우에는 실행되지 않음
  - ✓ 이름이 이미 목록에 존재하는 경우
  - ✓ 성별에 "남" 또는 "여" 외의 값이 입력된 경우
  - ✓ 중간고사와 기말고사에 0~100 외의 값이 입력된 경우

```
Command: add
Name: Alice → 이름이 존재하는 경우

Command: add
Name: 최아무개
Phone number: 01055667788
Gender: 나 → 남/여 외의 값이 입력된 경우

Command: add
Name: 최아무개
Phone number: 01055667788
Gender: 남
Midterm score: 106 → 0~100 외의 값이 입력된 경우
```

```
Command: add
Name: 최아무개
Phone number: 01055667788
Gender: 남
Midterm score: 93
Final score: 95
Student added.
```

```
Command: add
Name: Aaron
Phone number: 01011223344
Gender: 남
Midterm score: 66
Final score: 51
Student added.
```

```
Command: show
목록
```

	Name	Phone Number	Gender	Midterm	Final	Grade
1.	Aaron	01011223344	남	66	51	F
2.	Alice	0541236543	여	76	89	B
3.	Bob	0312254567	남	58	68	D
4.	김철수	01098761234	남	57	62	F
5.	이영희	021112222	여	81	84	B
6.	최아무개	01055667788	남	93	95	A
7.	홍길동	01012345678	남	84	73	C

```
Command:
```

## 7. 학생 삭제 (delete)

- **delete** 입력 시, 삭제하고자 하는 학생의 이름을 입력 받아 해당하는 학생의 항목을 삭제한다. 삭제하면, 메시지 "Student deleted."를 아래와 같이 출력한다.
- 학생 삭제 후 **show** 명령어를 사용하면 영어(알파벳 순) -> 한글(가나다 순) 순으로 정렬된 상태를 유지한다.

- 예외처리:

- ✓ 목록에 아무도 없을 경우 아래의 예제와 같이 "List is empty." 메시지 출력

```
Command: delete
List is empty.
```

```
Command:
```

- ✓ 삭제하려는 이름이 존재하지 않을 경우 실행되지 않음

```
Command: delete
Name to delete: Bobby → 이름이 없는 경우

Command: delete
Name to delete: Bob
Student deleted.

Command: show
목록

   Name      | Phone Number | Gender | Midterm | Final | Grade
1. Aaron    | 01011223344 |   M   |    66   |    51   |   F
2. Alice    | 0541236543  |   F   |    76   |    89   |   B
3. 김철수   | 01098761234 |   M   |    57   |    62   |   F
4. 이영희   | 021112222  |   F   |    81   |    84   |   B
5. 최아무개 | 01055667788 |   M   |    93   |    95   |   A
6. 홍길동   | 01012345678 |   M   |    84   |    73   |   C

Command: _
```

## 8. 종료 (quit)

- **quit** 입력 시, 프로그램을 종료한다.
- 해당 명령어를 실행할 경우, 현재까지 편집할 내용의 저장 여부를 묻고, 저장을 선택할 경우 파일명을 입력 받아서 저장하도록 한다. 앞서 본 "students.txt"와 같이 내용을 구성한다.

[Name][Wt][PhoneNumber][Wt][Gender][Wt][Midterm][Wt][Final][Wn]

- 저장할 때 목록의 순서는 영어(알파벳 순) -> 한글(가나다 순) 순으로 한다.
- 파일 이름에는 공백이 없다고 가정한다.
- 동적 할당된 memory 를 모두 free 시킨 후 종료한다.

```
Command: quit
Save data?(y/n) y
File name: new_students.txt
계속하려면 아무 키나 누르십시오 . . .
```